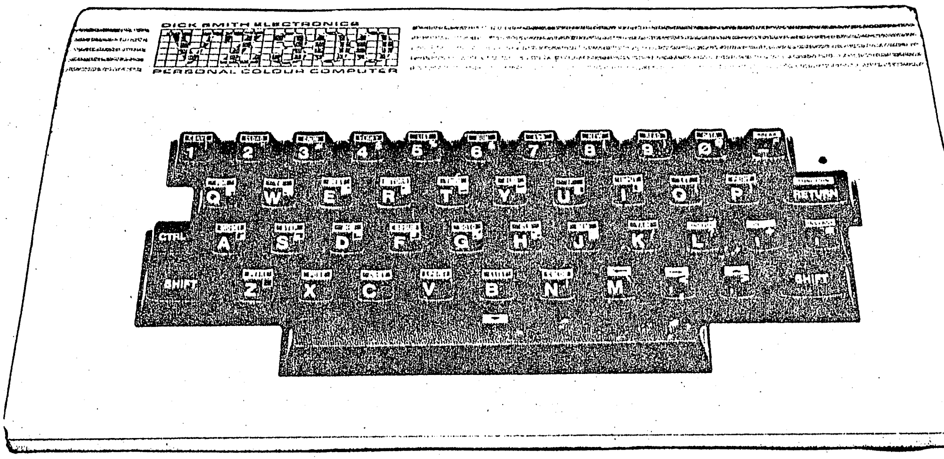
MARCH/APRIL 1987 ISSUE # 10/11

HUNTER VALLEY VZ JOURNAL



PRODUCED BY H.V.V.Z.U.G. . .

Carrying on from Jan/Feb issue the Hunter Valley VZ Journal will be published Bi-Monthly for the forseeable future. This will bring us in line with other VZ Publications and also give us more time to produce a better quality Journal.

BYTE BACK & OTHER VZ PUBLICATIONS :- Page 3
Our list of VZ User Groups and Publications is growing.

SNOOPY CALENDAR PART 2 by Dave Boyce :- Pages 4-5
This concludes listing of Snoopy Calendar. The lines of hashes are simply a guide to help you when you are typing in.

PRINTER/PLOTER SECTION by Dave Boyce :- Page 6
Dave shows you how to convert Snoopy Calendar for Printer/Plotter use.

DISK DRIVE HINTS & TIPS :- Pages 5 & 10

RAM COMMUNICATIONS AREA by David Mitchell (C) Pages 7-8
Dave has put in a lot of time and effort in compiling the list and a lot of VZ users will appreciate his sharing the result of his labours. Thanks Dave. There are a few gaps in the list and we would like your help in filling them.

ROM INVERSE CHARACTER SET by Bob KITCH (C) Pages 9-10
Bob's article explains in his usual thorough way how inverse characters are stored in ROM an why they wont print out on most Dot Matrix Printers. A screen dump is shown at end of listing.

The MAGIC VZ - UTILITY REVIEW :- Pages 11-12
The next best thing to a disk drive when you don't have a disk drive or how to live without one when you can't afford one let alone two or three.

UTILITY REVIEW by Larry Taylor (C) :- Page 13
FILESEARCH is a utility that can be used both by tape or disk drive owners. It gives quite a lot of information on your programs. This is Larry's first contribution to our Journal and we welcome him aboard.

CONVERTING TAPE MAILING LIST FOR DISK USE :- Page 14
D.Smith has available quite a few Technical Bulletins for the VZ. Bulletin 111 is reproduced by kind permission D.Smith Computer User Support.

BEGINNER'S SECTION by Leigh Rogers :- Pages 15-16
In her easy to follow style Leigh takes you line by line explaining as she goes along.

ENHANCING VZ BASIC by Larry Taylor (C) :- Pages 17-19
This issue has our our first assembly listing. The basic version will appear in next issue and intending assembly learners will find it helpfull.

VIDEO DISPLAY WORKSHEET (MODE 0) :- Page 20

BELIEVE IT OR NOT :-
Two female computers were overheard gossiping. (Via their speech synthesisers - Ed.). One asked the other. "How did your honeymoon go" ? "terrible", replied the other, "He left his Joystick at home".

Since our comments last month we had a few NIBBLES. Here are some of the comments.

Dave Boyce - It is nice to receive a Mag. (sorry Journal) that is pages thick. Dave also suggested that we do an annual index together with notes and errata. Thanks for the suggestion Dave, we had something similiar in mind.

Scott Le Brun - I've been reading the Jan/Feb issue of your Journal and IT'S GREAT. Thanks Scott. The RESTORE article, I found very useful, as I sometimes accidentally NEW a program and it's not stored on tape. Take a bow Dave Mitchell.

With this issue we are introducing a new service to our members on a trial basis. Most programs appearing in this Journal can be obtained from Hunter Valley VZ Users' Group simply by sending a TAPE or DISK together with return POSTAGE and PACKING included. When sending disks make sure you use stiff cardboard as support as the postman likes folding disks in half to fit in your letter box. 'P' plates are ideal as support.

OTHER VZ PUBLICATIONS, GROUPS . .

VZ USER
MARK HARWOOD
P.O. BOX 154
DURAL 2158
N.S.W.

$15.00 PER ANUM

LE'VZ 200/300 OOP
J.C.E. D'ALTON
39 AGNES STreet
TOOWONG 4066
QUEENSLAND

$2.00 PER ISSUE (min $4.00)
VSOFTWAREZ
Software/Hardware for sale.

VZ DOWN UNDER
C:- GEORGE SEGGIE
P.O. BOX 316
ST. KILDA 3182
VICTORIA

Software/Hardware for sale

MICRO-MAGIC
GORDON BROWELL
13 BROOKES St.
BIGGENDEN 4621
QUEENSLAND

NOTE :- When writing to any above or H.V.VZ. Users' Group for information please enclose a S.S.A.E. (Stamped Self Addressed Envelope).

HUNTER VALLEY VZ USERS' GROUP . .

PETER ELLIS . . . . PRESIDENT . . . (049) 69 5697
MARK O'BRIEN . . . . . VICE PRESIDENT . (049) 61 5490
ROSS WOODS . . . . SECRETARY . . . . (049) 71 2843
LEIGH ROGERS . . . . . . TREASURER . . . . (049) 57 5738
JOE LEON . . . . EDITOR . . . . (049) 51 2756

MAIL ALL SUBMISSIONS TO :-
HUNTER VALLEY VZ USERS' GROUP
C/- P.O. BOX 161 JESMOND
N.S.W. 2299

```
840 LPRINT
845 '----5----5---X#####X#####X#####X#####X#####X#####X#####X#####X##X
850 LPRINTTAB(30);"             XXXX"
860 LPRINTTAB(30);"            X    XX"
870 LPRINTTAB(30);"           X  ***  X                XXXXX"
880 LPRINTTAB(30);"          X  *****   X            XX     XX"
890 LPRINTTAB(30);"      XXXXX *******  XXX      XXXX         ";
892 LPRINT"XX"
900 LPRINTTAB(30);"    XX    X ******   XXXXXXXXX             ";
902 LPRINT"  XX XX"
920 LPRINTTAB(30);"   XX       X ***    X                     ";
922 LPRINT"   X** X"
940 LPRINTTAB(30);"  X         XX    XX      X                ";
942 LPRINT"   X***X"
960 LPRINTTAB(30);" X          //XXXX        X                ";
962 LPRINT"   XXXX"
980 LPRINTTAB(30);"X          //    X                         ";
982 LPRINT" XX"
985 '----5----5---X#####X#####X#####X#####X#####X#####X#####X#####X##X
1000 LPRINTTAB(30);"X         //     X          XXXXXXXXXXXXXX";
1002 LPRINT"XX/"
1020 LPRINTTAB(30);"X     XXX//     X          X"
1040 LPRINTTAB(30);"X    X   X      X         X"
1050 LPRINTTAB(30);"X    X    X     X        X"
1060 LPRINTTAB(30);"X    X    X     X        X                ";
1062 LPRINT"   XX"
1070 LPRINTTAB(30);" X    X   X     X        X                ";
1072 LPRINT"XXX  XX"
1080 LPRINTTAB(30);" X     XXX       X       X               X";
1082 LPRINT"  X X  X"
1090 LPRINTTAB(30);" X              X         X              X";
1092 LPRINT"X X  XXXX"
1100 LPRINTTAB(30);"  X              X         XXXXXXX\      X";
1102 LPRINT"X   XX  X"
1110 LPRINTTAB(30);"   XX            XX              X       X";
1112 LPRINT"    X  XX"
1120 LPRINTTAB(30);"     XX            XXXX   XXXXXX/       X ";
1122 LPRINT"    XXXX"
1125 '----5----5---X#####X#####X#####X#####X#####X#####X#####X#####X#X
1130 LPRINTTAB(30);"       XXX             XXX***          X  ";
1132 LPRINT"  X"
1140 LPRINTTAB(30);"        XXXXXXXXXXXXXXXX *   *        X   ";
1142 LPRINT" X"
1150 LPRINTTAB(30);"                        *---* X      X    ";
1152 LPRINT"X"
1160 LPRINTTAB(30);"                       *-* *   XXXX X    X"
1170 LPRINTTAB(30);"                       *- *       XXX   X"
1180 LPRINTTAB(30);"                      *- *X          XXX"
1190 LPRINTTAB(30);"                      *- *X  X          XXX"
1200 LPRINTTAB(30);"                     *- *X    X           ";
1202 LPRINT" XX"
1210 LPRINTTAB(30);"                     *- *XX    X          ";
1212 LPRINT"   X"
1220 LPRINTTAB(30);"                    *  *X* X    X         ";
1222 LPRINT"    X"
1230 LPRINTTAB(30);"                    *  *X * X    X        ";
1232 LPRINT"     X"
1234 '-----5-----5---X#####X#####X#####X#####X#####X#####X#####X#####X#X
```

```
1236 '----5----5---X#####X#####X#####X#####X#####X#####X#####X#X
1240 LPRINTTAB(30);"                    *  * X**   X   XXXX     ";
1242 LPRINT"     X"
1250 LPRINTTAB(30);"                    *  * X**  XX      X      ";
1252 LPRINT"     X"
1260 LPRINTTAB(30);"                   *  ** X** X      XX      ";
1262 LPRINT"    X"
1270 LPRINTTAB(30);"                   *  **  X*  XXX     X      ";
1272 LPRINT"   X"
1280 LPRINTTAB(30);"                  *  **     XX     XXX        X";
1282 LPRINT"XX"
1290 LPRINTTAB(30);"                 *  * *      XXXX          X   ";
1291 '----5----5---X#####X#####X#####X#####X#####X#####X#####X#X
1292 LPRINT" X"
1300 LPRINTTAB(30);"                *  * **            X     X   ";
1302 LPRINT" X"
1310 LPRINTTAB(30);" =======*******  *  *           X     X    ";
1312 LPRINT"  XXXXXXXXX\"
1320 LPRINTTAB(30);"        *        * *       /XXXXXX     XXX";
1322 LPRINT"XXXX\       )"
1330 LPRINTTAB(30);"    ====******** * *      X                ";
1332 LPRINT"      )\      )"
1340 LPRINTTAB(30);"      ===*         *     X               \";
1342 LPRINT"   \ )XXXXXX"
1350 LPRINTTAB(30);"=========**********       XXXXXXXXXXXXXXXX";
1352 LPRINT"XXXXX"
1355 '----5----5---X#####X#####X#####X#####X#####X#####X#####X#X
1360 LPRINT
1380 RETURN
1390 END
1500 ERA"SNOOPYCL"
1600 SAVE"SNOOPYCL":CLS:DIR
```

DISK DRIVE HINTS AND TIPS . . . .

Hints for the VZ abound while ones for the disk drive are few and far between. Here are some you may find usefull.

How many times have you Run a program without SAVING it first and lost it all because the VZ or drive hung up. Fear not, there is a simple solution. Just add a couple extra lines to your program. The trick is to SAVE your program regularly.

```
1500 ERA"FILENAME"
1600 SAVE"FILENAME":CLS:DIR
```

The first time you save your program you would use line 1600 :-

```
GOTO 1600
```

This will SAVE your program, clear the screen and show you the DIRECTORY. Then at regular intervals type in :-

```
GOTO 1500
```

Provided you a have disk in the drive line 1500 will ERASE your program while line 1600 will SAVE an updated version of your program.

Instead of line 1500 and 1600 any line number of your choice can be used. Have a look at lines 1500 and 1600 of Snoopy Calendar program in this issue for an example.

CHANGING SNOOPY CALENDAR FOR PRINTER/PLOTTER USE by Dave Boyce.

Now that You have typed in Part 2 of the Calendar we can procede. Load in the completed program and follow the instructions given below.

Delete Lines 162, 348, 558 and 820.

Type in -: SAVE"SNOCALPP" and press RETURN. (DISK SAVE)

Type in -: CSAVE"SNOOPY/CAL" and press RETURN. (TAPE SAVE)

Add Lines 70, 80, 160, 285, 595 and 1255 as Listed below.

Change Lines 90 to 120, 345, 365, 375, 385, 552, 580, 585, 590, 830 and 1500 as Listed below.

Also change the TAB(30)'s in Lines 850 to 1350 to TAB(15)'s as the example shows below.

Type in -: GOTO 1500 and press RETURN. (DISK SAVE)

Type in -: GOTO 1600 and press RETURN. (TAPE SAVE)

NOTE :- In case of DISK SAVE use line 1500 and line 1600 for TAPE SAVE whenever insructions call for program SAVE.

```
70 '  FILE - SNOCALPP
80 '  ##  SNOOPY CALENDAR ##
90 '  FOR THE PRINTER/PLOTTER ONLY
100 ' ## LINE 160 WILL SET YOUR PRINTER/PLOTTER
105 ' IN THE CORRECT MODE
110 ' ### RUN LINE 595 TO RETURN TO TEXT MODE
120 ' IF LEAVING PROGRAM PREMATURELY
160 LPRINTCHR$(18):LPRINT"SO":LPRINTCHR$(17)
285 LPRINT
345 LPRINT:LPRINT:LPRINT
365 LPRINTB$
375 LPRINT:LPRINT
385 LPRINT:LPRINTA$
552 GOSUB830:' SOOPY PRINTOUT
580 CLS:INPUT"ANOTHER YEAR ";YY$:IFYY$="Y"THEN RUN ELSE 595
585 END
590 ' ### RETURN TO TEXT MODE
595 LPRINTCHR$(18):LPRINT"S1":LPRINTCHR$(17)
830 ' SNOOPY PRINTOUT
845 '----5----5---X#####X#####X#####X#####X#####X#####X#####X##X
850 LPRINTTAB(15);"                    XXXX"
860 LPRINTTAB(15);"                  X     XX"
1255 '----5----5---X#####X#####X#####X#####X#####X#####X#####X#X
1400 :
1450 :
1500 ERA"SNOCALPP":SAVE"SNOCALPP":DIR:' THIS LINE FOR DISK ONLY
1550 :
1560 :
1600 CSAVE"SNOOPY/CAL":'TAPE USERS USE THIS LINE TO SAVE PROGRAM
```

BY DAVID MITCHELL

30720 - 30740 7800 - 7814 RST JUMP VECTORS

30720-2	7800-2	RST 8 JP 1C96
30723-5	7803-5	RST 10 JP 1D78 (DOS 4293)
30726-8	7806-8	RST 18 JP 1C90
30729-31	7809-B	RST 20 JP 25D9
30732-4	780C-E	RST 28 SET TO RETURN
30735-7	780F-11	RST 30 SET TO RETURN
30738-40	7812-4	RST 38 SET TO EI,RETURN

30741 - 30748 7815 - 781C KEYBOARD D.B.C.

30741	7815	DEVICE TYPE
30742-3	7816	DRIVER ADDRESS
30744	7818	INVERSE SCREEN
30745	7819	INVERSE TYPING
30746	781A	?
30747-8	781B-C	RAM BUFFER ADDRESS

30749 - 30756 781D - 7824 VIDEO D.B.C.

30749	781D	DEVICE TYPE
30750-1	781E-F	DRIVER ADDRESS
30752-3	7820-1	CURSOR POSITION
30754	7822	?
30755-6	7823-4	CHECKSUM DURING TAPE LOAD

30757 - 30764 7825 - 782C PRINTER D.B.C.

30757	7825	DEVICE TYPE
30758-9	7826-7	DRIVER ADDRESS
30760	7828	LINES PER PAGE
30761	7829	LINES PRINTED SO FAR
30762	782A	?
30763-4	782B-C	RAM BUFFER ADDRESS

30765 - 31058 782D - 7952 COMMUNICATION REGION

30765-73	782D-35	USED BY TRS-80 DOS
30774-5	7836-7	?
30776	7838	INVERSE MESSAGE
30777	7839	DISABLE UP,DOWN KEYS:LOW RES SPEED UP.
30778	783A	?
30779	783B	COPY OF O/P LATCH
30780	783C	?
30781	783D	VIDEO CONTROL WORD
30782-844	783E-7C	?
30845-7	787D-F	INTERRUPT
30848-61	7880-D	DIVISION SUPPORT ROUTINE
30862-3	788E-F	USR POINTER
30864-6	7890-2	RND NUMBER SEED 30867-9 7893-5 INP (XX)
30870-2	7896-8	OUT (XX)
30873	7899	LAST KEY PRESSED
30874	789A	ERROR CODE STORAGE
30875	789B	PRINTER CARRIAGE POSITION
30876	789C	OUTPUT DEVICE CODE
30877	789D	SIZE OF VIDEO LINE
30878	789E	HIGH OR LOW RES 30879 789F ?
30880-1	78A0-1	BASIC STACK ADDRESS
30882-3	78A2-3	BASIC LINE NUMBER
30884-5	78A4-5	START OF PROGRAM
30886	78A6	TAB CURSOR POSITION

30887-8	78A7-8	KEYBOARD BUFFER ADDRESS
30889	78A9	?
30890-3	78AA-D	RANDOM NUMBER SEED
30894	78AE	VARIABLE FLAG
30895	78AF	NUMBER FLAG (INTEGER,STRING,SINGLE,DOUBLE)
30896	78B0	EXPRESSION EVALUATION
30897-8	78B1-2	TOP OF MEMORY
30899-900	78B3-4	NEXT LOCATION IN LITERAL STRING POOL
30901-2	78B5-6	START OF LITERAL STRING POOL
30903-45	78B7-E1	?
30946-7	78E2-3	CURRENT LINE NUMBER
30948-9	78E4-5	AUTO INCREMENT
30950-1	78E6-7	LAST BYTE EXECUTED IN CURRENT LINE
30952-3	78E8-9	BACKSPACED STACK ADDRESS
30954-5	78EA-B	ERROR LINE NUMBER
30956-7	78EC-D	ERROR LINE NUMBER
30958-9	78EE-F	CURSOR POSITION IN LINE WITH ERROR
30960-1	78F0-1	ADDRESS OF STATEMENT TO RESUME AT
30962	78F2	ERROR MESSAGE OVERRIDE
30963-4	78F3-4	?
30965-6	78F5-6	LINE NUMBER ENDED ON
30967-8	78F7-8	LAST STATMENT BYTE SCANNED
30969-70	78F9-A	END OF PROGRAM
30971-2	78FB-C	START OF DIM VARIABLES TABLE
30973-4	78FD-E	START OF FREE MEMORY
30975-6	78FF-900	DATA POINTER (READ STATEMENT)
30977-1002	7901-1A	VARIABLE DECLARATION TABLE
31003	791B	TRACE FLAG
31004	791C	MATH ROUTINES

31005 - 31012 791D - 7924 regester 1

		INTEGER	SINGLE	DOUBLE
31005	791D			LSB
31006	791E			LSB
31007	791F			LSB
31008	7920			LSB
31009	7921	LSB	LSB	LSB
31010	7922	MSB	LSB	LSB
31011	7923		MSB	MSB
31012	7924		EXP	EXP

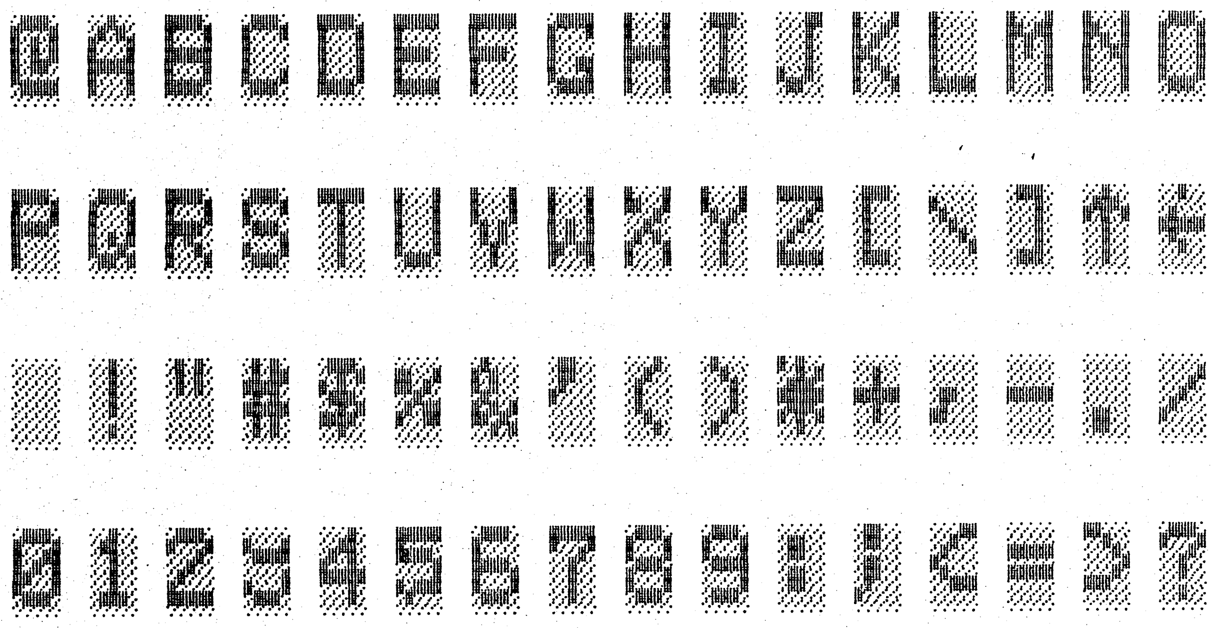
31013-4	7925-6	MATH ROUTINES
31015-22	7927-E	REG 2 SAME AS REG 1
31023	792F	?
31024-49	7930-49	PRINTER BUFFER
31050-7	794A-51	DOUBLE PRECISSION MATH
31058 - 31141	7952 - 79A5	TRS-80 DOS RESERVED WORDS
31141 - 31204	79A6 - 79E4	TRS-80 DOS LINKS
31205-7	79E5-7	USED BY INPUT
31208-72	79E8-7A28	BASIC LINE INPUT BUFFER
31273 - 31388	7A29 - 7A9C	?
31389 - 31404	7A9D - 7AAC	TAPE NAME
31405 - 31446	7AAD - 7AD6	?
31447 - 31462	7AD7 - 7AE6	TAPE LOAD ROUTINE
31463-4	7AE7-8	ZERO

31465	7AE9	START OF USER MEMORY
36864	9000	TOP OF MEMORY VZ-200 (6K)
45184	B800	TOP OF MEMORY VZ-300 (16K)
53348	D000	TOP OF MEMORY VZ-200 (16K)
63488	F800	TOP OF MEMORY VZ-300 (16K+16K)
65535	FFFF	THE VERY TOP

BY BOB KITCH (C) COPYRIGHT

```
10 '           ***********************************
20 '           ***   DISPLAY INVERSE CHARACTER   ***
30 '           ***          SET IN ROM           ***
40 '           ***     AS USED BY DOT MATRIX     ***
50 '           ***            PRINTER            ***
60 '           ***     R. B. KITCH   27/9/86     ***
70 '           ***********************************
80 '
100 'WHEN INVERSE CHARACTERS ARE SENT TO A DOT MATRIX PRINTER
110 'THE PRINTER SHIFTS TO GRAPHICS MODE AND REQUIRES A ROUTINE
120 'TO SUPPLY THE APPROPRIATE SHAPES TO THE HEAD. (NORMAL
130 'CHARACTERS ARE HELD IN THE PRINTERS ROM)
140 'IN THE VZ COMPUTER A TABLE OF SHAPES IS LOCATED AT
150 '3B94H TO 3CD3 IN ROM. THERE ARE 64 CHARACTERS, EACH USING
160 '5 BYTES TO DEFINE THEIR GRAPHIC SHAPE. THE SHAPES MAY BE
170 'DECODED AND OUTPUT TO THE SCREEN AS IS DONE IN THIS
180 'PROGRAM. NOTE THAT THERE ARE SOME ERRORS IN THE ROM.
190 'THE 5 BYTES DEFINE A 5 BY 8 DOT MATRIX WHICH IS THE SHAPE
200 'OF THE CHARACTER, WHICH INCIDENTLY ARE NOT ORDERED
210 'ACCORDING TO THE ASCII CODE.
220 'THE FIRST BYTE DEFINES THE LEFT HAND EDGE OF THE CHARACTER-
230 'WHICH IS THE FIRST PRINTED DURING A PASS OF THE PRINTER
240 'HEAD. IN TANDY PRINTERS THE MSB IS THE LOWERMOST PIN OF THE
250 'HEAD AND THE LSB IS THE UPPERMOST PIN. THE PINS ON EPSON
260 'PRINTER HEADS ARE ARRANGED IN THE OPPOSITE SENSE. THIS
270 'REQUIRES THAT THE BITS IN EACH BYTE BE REVERSED.
280 '************************************************************
290 '
300 DIM MK%(7) :'***VECTOR OF BIT MASK VALUES - POWERS OF 2
310 DIM BT%(7) :'***VECTOR OF DECODED BITS FROM ROM VALUE.
320 '
330 '***FILL MASK VECTOR WITH POWERS OF 2 FOR DECODING.
340 FOR I%=0 TO 7 :MK%(I%)=2^I% :NEXT I%
350 '
400 '***INITIALIZE PARAMETERS - MAY BE CHANGED TO VARY SCREEN.
410 CC%=4     :'***CHARACTER COLOUR.  (1-4)
420 BC%=2     :'***BACKGROUND COLOUR. (1-4)
430 CS%=0     :'***COLOUR SET. (0-1)
440 CW%=3     :'***COLUMN WIDTH BETWEEN CHARACTERS.
450 SP%=16    :'***ROW SPACING FOR CHARACTERS.
460 HS%=0     :'***STARTING HORIZONTAL POSITION ON HI-RES SCREEN.
470 VP%=3     :'***STARTING VERTICAL POSITION ON HI-RES SCREEN.
480 HM%=127   :'***MAXIMUM HORIZONTAL POSITION. (0-127)
490 '
600 '***SET UP MAIN LOOP TO STEP THROUGH ROM FROM 3B94H-3CD3.
610 BK%=0                    :'***BYTE COUNTER FOR EACH CHARACTER.
620 HP%=HS%                  :'***SET HORIZONTAL POSITION TO START
630 MODE(1) :COLOR,CS%       :'***SET HI-RES SCREEN AND COLOR SET.
640 FOR AD%=15252 TO 15571 :'***ROM ADDRESSES FOR SHAPE TABLE.
650   DV%=PEEK(AD%)          :'***DECIMAL VALUE IN ROM.
660 '
700 '***DECODE THE INDIVIDUAL BITS OF DV% AND STORE IN BT%().
710 '***THE MASK VALUES IN MK%() ARE "ANDED" WITH THE VALUE.
720 '***THE RESULT STORED IN BT%() IS THE "COLOUR" OF THE BIT.
730   FOR I%= 0 TO 7          :'***PROCEED FROM LSB TO MSB.
740     IF DV% AND MK%(I%) THEN BT%(I%)=BC% ELSE BT%(I%)=CC%
750   NEXT I%
800 '
810 '***CHECK THAT THERE IS ENOUGH ROOM TO PLOT CHARACTER.
820   IF BK%=0 AND HM%-HP%<4 THEN HP%=HS% :VP%=VP%+SP%:'*NEW ROW
830   BK%=BK%+1              :'***INCREMENT BYTE COUNTER.
```

```
840 '
900 '***OUTPUT BYTE TO SCREEN.
910   FOR I%=0 TO 7
920     COLOR BT%(I%)       :'***SET COLOUR OF BIT.
930     SET(HP%,VP%+I%)     :'***PLOT BIT.
940   NEXT I%
950 '
1000 '***PREPARE FOR NEXT BYTE.
1010   HP%=HP%+1             :'***INCREMENT HORIZONTAL POSITION.
1020   IF BK%=5 THEN BK%=0 :HP%=HP%+CW%       :'***NEW CHARACTER.
1030 NEXT AD%
2000 GOTO 2000 :END
```



DISK DRIVE HINTS AND TIPS - - - -

Most Dos commands can be used from within a program, but the DOS manual fails to inform you of the correct syntax. EG :-

100 IFA$="D"THENDIR : This will not work although it's correct for BASIC, but not for the DOS.

100 IFA$="D"THEN:DIR : A colon (:) must be added between THEN and DIR for it to work. The same goes for most other DOS commands.

BREAK :- Although there's a BREAK function for basic the DOS Manual again fails to inform you of the DOS version. It could'nt be simpler. Just use the BREAK (-) MINUS key by itself without CTRL key. It will get you out of some problems but not all.

Anyone with any hints for the disk drive please let us know so we can share with other users.

Disk Drive owners, the Magic VZ is not for you, but tape owners rejoice as you can do Magic with your VZ. And how to accomplish this Magic ? Simple, just install a Magic Eprom in your VZ200/300 and away you go. The Magic Eprom is a 4K Eprom that is written for the range 4000-4FFF HEX. It contains the following three utilities which are available seperately on tape. 1) EB1 - 2) MDR3 - 3) AR1 :-

EB1 - EXTENDED BASIC :-
All 45 missing TRS-80 Level 2 Keywords are translated into their Tokens and are listable any time. This gives you 19 additional commands/functions and all the disk command exits which are very handy to link any M/L programs. Note that basic programs written with EB1 will run on any other VZ even if commands/functions from the list below are used.

ADDITIONAL COMMANDS/FUNCTIONS :-
POS, FRE(X$), MEM, ERR, ERL, FIX, CDBL, CSNG, CINT, ON, AUTO, DEFINT, DEFSNG, DEFDBL, DELETE, RANDOM, VARPTR, STRING$, RESUME.

DISK COMMANDS/FUNCTIONS :-
LOF, LOC, EOF, CVD, CVS, CVI, DEF, PUT, GET, MKD$, MKS$, MKI$, TIME$, CMD, FN, LSET, RSET, OPEN, CLOSE, LOAD, SAVE, KILL, NAME, MERGE, FIELD, INSTR.

Information is given on how to realise ON ERROR, TRON, TROFF and DEFSTR.

MDR3 - MERGE/DELETE/RENUMBER :-
The merge routine will merge any number of basic programs up to memory capacity. The only rule is that the program you wish to merge must have it's first line number higher than the last line of program in memory.

REN X,Z or REN X-Y,Z.
This is a particulary usefull RENUMBER routine. You can renumber whole or part of your program with choice of increment.

DELETE X/X-Y/-X :- The specified Line(s) are deleted.

AR1 - ARRAY-UTILITY/RESTORE N :-
This powerfull utility can dump 800 records of 16 Characters in less than 4 minutes onto tape. With this ARRAY Utility you can SAVE or LOAD whole ARRAYS at the tape speed of 600 baud, regardless of the lenght or the number of dimensions. RESTORE N enables you to move the DATA pointer for the READ command to the first element of any DATA line in your program. Important, the array utility will only work if the basic program was written by an extended basic which accepts the disk commands LOAD, SAVE, KILL and NAME.

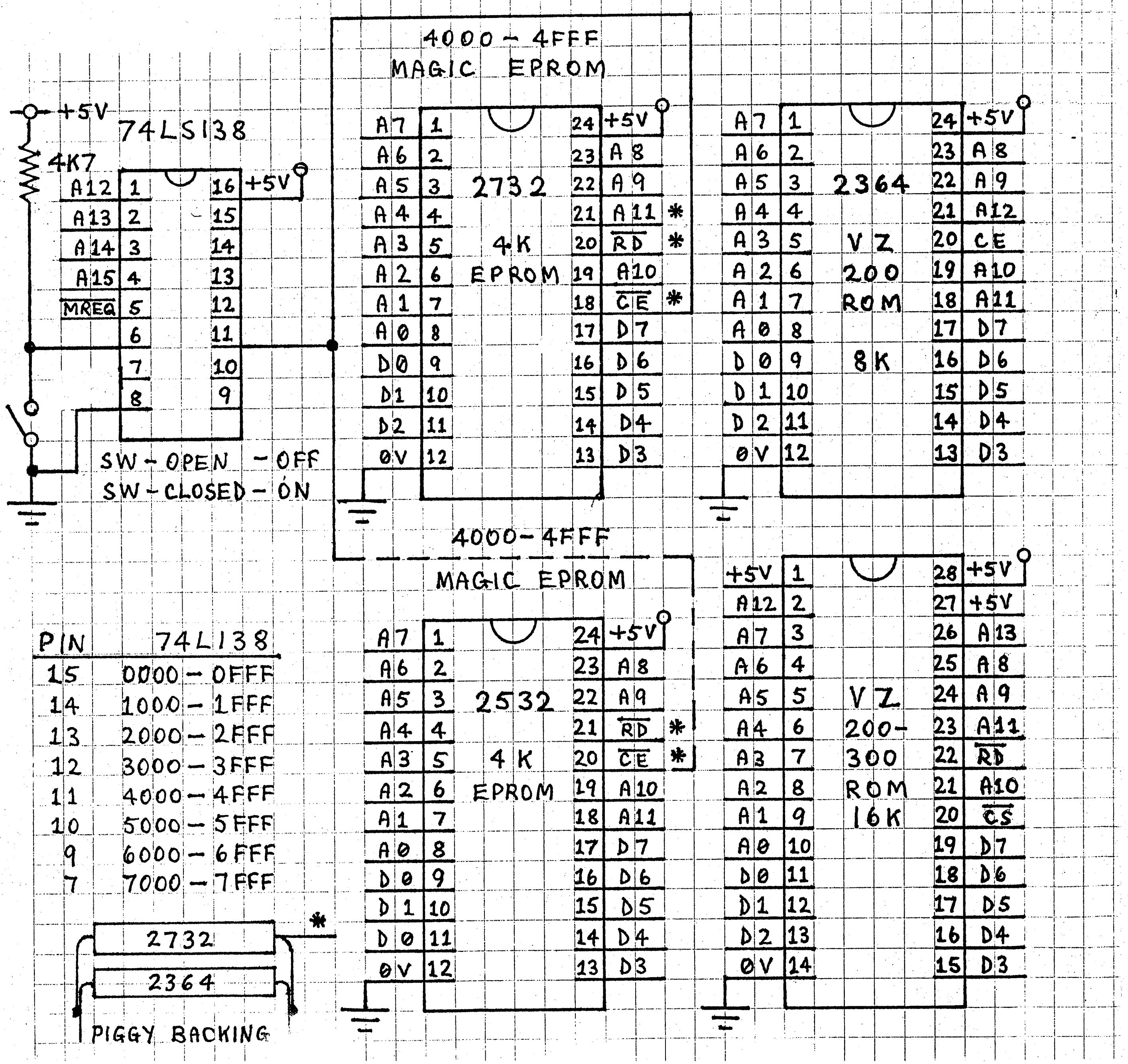
IMPORTANT NOTE :-
EB1 can be loaded with either MDR3 or AR1. They can be loaded any time without destroying a BASIC program.
EB1, MDR3, AR1 and MAGIC EPROM are available from :-

W.Obrist	EB1, MDR3 and AR1 are $15.00 each.
50 Cobham Ave.	MAGIC EPROM is $45.00.
West Ryde	They are last year's prices so check.
N.S.W. 2114	State RAM size. EG :- 8K, 24 K, ETC.

MAGIC-EPROM :- This Eprom contains all three utilities described previously EB1, MDR3 and AR1. When correctly installed all pointers are changed during power up resulting in a very powerfull VZ. The beauty of the Magic Eprom is that it works with any memory size and all commands and functions are available at power up.

The Eprom is available as 2532 or 2732, and don't forget to mention your choice. To make your choice easier pinouts and connecting circuit diagrams are shown. I piggy backed mine on a 2364 8K-ROM inside the VZ200. Some VZ200's and VZ300's have 16K ROMS and their pinouts are also shown.

Both the 2532 and 2732 Eproms will fit nicely on the 2364 Rom in VZ200. Astericks are shown which pins to bend up when piggy-backing on 2364 VZ Rom. On the 2532 two pins must be bent up while on 2732 three are bent up and connected to the appropriate signals. Instead of soldering the Eprom directly to the Rom a socket can be used. If piggy-backing on the 16K Rom then the 2732 would be a better choice as only pins 18 and 24 need bending up. Study the circuits carefully before proceeding and take care.



FILESEARCH 2.0 (C)1987

===

===

FILESEARCH is a utility program, consisting of two independent modules TAPESEARCH and DISKSEARCH, which provides useful information about programs stored on tape or disk. Although the data appears on the screen, selection of the printer option enables a printout to be obtained.

The initial letter identifies the type of program:

T: Basic Program
B: Machine Code Program
D: Data Files
W: Editor Assembler Source Code

TAPESEARCH reads the leader of programs and files stored on tape and prints the information in this form:

T:NAME OF PROGRAM 7AE9 8EA4 13BB

Following the program type and name, the start and end addresses and the length of the program will be printed, in hex. Only Data Files do not have a start and end address.

TAPESEARCH will also identify Data Files saved using the D.S.E. Word Processor, which cannot be identified during normal tape operations. When the printer option is selected only Data Files which have a name are output to the printer although all files found are printed on the screen.

DISKSEARCH reads the disk directory, and prints the data in this form:

T:FILENAME 0A 04 7AE9 8EA4 13BB

Following the filetype and filename the track and sector numbers, indicating where the data is stored on the disk, are displayed in hex form. Immediately after this the start and end addresses and the length of the program are printed, also in hex.

If the directory contains more than 13 entries, as the data is being printed, the screen will scroll. Pressing the SPACE bar will hold the displayed data. If the printer option has been chosen, you can select, each time a directory is displayed, whether to output the data to the printer or not.

If a disk error occurs during reading, control will return to DISKSEARCH. Should the program not return from reading a disk, such as if an attempt is made to read an unformatted disk, control can be regained by pressing the BREAK key alone.

FILESEARCH can be safely used by non-disk owners as a check is made for the presence of the Disk Operating System before entering DISKSEARCH. Pressing the CONTROL and BREAK keys together, at any time, will enable a return to the start of the program.

===

VZ-300 MAILING LIST TAPE TO DISK FILE CONVERSION

Below are the changes to be done to the B.A.S.I.C. program to allow files to be saved on disk instead of tape for (X-7259) Mailing List program.

Once you have LOADed Mailing List BREAK the program and type in the lines below pressing (RETURN) after each line.

```
1040 PRINT@162,"2. READ DATA FROM DISK";
1080 PRINT@290,"6. WRITE DATA TO DISK";
5020 PRINT@270,"[WRITE DATA TO DISK]"
5030
5040
5050
5060
5070
5080
5110
5120
5205 ERA"MAILDATA"
5210 OPEN"MAILDATA",1:PR#"MAILDATA", DT
5230 PR#"MAILDATA",D$(N)
5240 NEXT:CLOSE"MAILDATA"
6020 PRINT@79,"[ READ DATA FROM DISK ]";
6030
6040
6050
6060
6070
6080
6100 OPEN"MAILDATA",0:IN#"MAILDATA", DT:IF DT=OTHEN 6135
6120 IN#"MAILDATA",D$(N)
6135 CLOSE"MAILDATA"
7030 PRINT@199,"[FUNCTION COMPLETE]";
7050 SOUND 30,2:RETURN
```

Now SAVE"MAILLIST" to disk. Type NEW then press RETURN.
Type in and RUN program below.

```
10 OPEN"MAILDATA",1
20 PR#"MAILDATA",0
30 CLOSE"MAILDATA"
```

The above program has prepared the disk with the MAILLIST program to SAVE and READ files. The above program will never be used again.

Now you have finished just RUN"MAILLIST" and the instructions are as per old Mailing List program. The only difference is that it SAVES and LOADS files a lot faster.

Compiled by Jamie PERRY

G'day! My last article, (which was also my first), seems to have hit the spot, so I'm back for a sequel ... (Anyone want to buy the Movie Rights ? - Ed.)

OK! You're in for a more complex one this time. This one covers a lot and demonstrates a lot of stuff. Like last time the program listing is given first, then each line and command is explained.

```
10 CLS
20 A=RND(100)
30 INPUT "GUESS MY NUMBER (0-100)";B
40 IF A>B INPUT "HIGHER";B
50 IF A<B INPUT "LOWER";B
60 IF A=B PRINT "THAT'S RIGHT":GOTO 80
70 GOTO 40
80 INPUT "TRY AGAIN (Y/N)";C$
90 IF C$="Y" GOTO 10
100 PRINT "TYPICAL HUMAN!":END
```

This is a simple little game where you can communicate with the computer. It will choose a number at random and you have to guess what the number is. You will be given clues like "HIGHER" or "LOWER". By looking at some of the features of this programme, you might be able to see how easy it is to write your own games.

10 CLS :Clears the screen.

20 A=RND(100)

This is a feature we have not looked at yet. It is a function which enables the computer to chose a number at random. For this, we just print "RND" and straight after, you state (in brackets), the highest number you wish to be chosen. So if you were choosing lotto numbers, it would be : RND(40). Get it? In line 20, we have labelled our random number "A" by saying that "A=RND(100)".

30 INPUT "GUESS MY NUMBER (0-100)";B

This is another function we have not yet tackled, but don't worry, it is not hard to understand. INPUT means that the computer is looking for an input from you. It has told you what sort of input it is looking for, by asking you to "GUESS MY NUMBER?". The PRINT and INPUT statements are very similar but there is one difference. The PRINT command simply prints a message on the screen, whereas the INPUT command both prints a message on the screen, and waits for a reaction from the user. Your reaction is then labelled with another variable : we have labelled the response "B". Don't forget the semi-colon (;), and there is no need to type in question marks on the VZ as it puts them up for you the moment it sees an INPUT statement.

40 IF A>B INPUT "HIGHER";B

This says exactly that : if the random number (A) is bigger than the response (B), then tell the user that it is higher, wait for another guess, and replace his or her last response (B) with the new response.

50 IF A<B INPUT "LOWER";B

Again, this is easy to understand : if the random number (A) is lower than the response (B), then tell the user that his or her guess is too high, wait for another response, and again, label it B.

60 IF A=B PRINT "THAT'S RIGHT":GOTO 80

This is the third thing that could possibly happen, he or she gets it right. It says that if his or her guess equals the random number, then print up that he or she is right. I have then redirected that part of the programme elsewhere by saying "GOTO 80".

70 GOTO 40

If you will notice, the majority of the programme will be running through lines 40, 50, and 60. Once the computer has been through these lines, it would be a very short game if the computer were not sent back through them to let you have another go ... So we say "GOTO 40", (in other words keep asking me, do I want another go).

80 INPUT "TRY AGAIN (Y/N)";C$

This is where line 60 sent the computer to. If you have guessed right, it asks you, "TRY AGAIN (Y/N)?". Again, it is waiting for a response and a simple thing like telling the user what sort of response to give, (Y/N) makes it all the easier for the user. You will notice that this response has been labelled "C$". This is because only numbers can be labelled with a single letter variable. If the response is going to be a letter, it must have a dollar sign ($) tacked on so the computer can distinguish between the two.

90 IF C$="Y" GOTO 10

If our response (C$) is "yes" (Y), start again at line 10. Notice that the response has been placed in "". This is also because it is a letter and not just a simple number. For some reason, a computer has no trouble dealing with numbers, (Only people - Ed.) but when letters are involved, the matter becomes a bit more complex.

100 PRINT "TYPICAL HUMAN!":END

I have put in a bit of cheek here. You must remember, that in line 90, there is a condition : "IF C$="Y" GOTO 10". Only if C$ is "Y", is that line obeyed. If the response is anything else, the computer will ignore the line and continue with the rest of the programme. Since it has been told to print up a message it will do so, which will probably surprise anybody else who sees it!. After the colon (:), it sees a fresh command which of course, tells it to finish up.

Anyway, that's it for this month. It is a long one but it should keep you going. See ya!

Enhancing VZ Basic by Larry Taylor

The Commodore 64 has advanced hardware supported by an inadequate Basic language, resulting in a number of enhanced Basics being available. Something similar could be produced for the VZ. It must be noted, however, that all such Basics share a common disadvantage. Any program which makes use of them requires the language be loaded before it will function properly.

Because Basic is an interpreted language additional commands can be inserted, if they can be intercepted and executed before reaching the VZ's own interpreter. This is precisely what happens when a disk operating system (DOS) is added. New commands enabling disk operations to be performed, supplement the existing Basic. However, all programs using those extra commands require the DOS to be present before execution or they will not be interpreted correctly.

When a Basic program is RUN, control passes to a machine language ROM routine, the Execution Driver at 1D5AH, which scans each line of the Basic program as it comes to it and begins to translate it. Part of the translation process involves looking for tokens. These are values in the range 128-250 (80H-FAH) that take the place of Basic reserved words e.g. CLS = 132 (84H). Once the word has been identified and checked for correct syntax, control is passed to the corresponding ROM routine before returning to continue the translation. This is similar to one person issuing instructions to another through an interpreter, who first has to translate them before the receiver can act, and is the reason for Basic's slow execution. Most languages get around this problem by having the program translated or compiled before execution.

Tandy's Colour Computer has an enchanced CLS command which enables the user to clear the screen to any one of nine background colours. The syntax is CLSn, where n may be a number in the range 0-8. To illustrate how enhancements can be accomplished, this command will be added to the VZ's repertoire.

On power up the address of the routine which examines each byte in a line of Basic, is stored at 7804H. Because this address is in RAM it can be easily changed. This was done so that at a later stage the DOS could be included. However, it also means that, just as readily, an enhanced form of Basic may be added. The trick is to ensure that, as far as the VZ's interpreter is concerned, nothing unusual has happened. The accompanying assembly language listing shows how this can be accomplished.

Having adjusted the top of memory pointer, the address at 7804H is stored and replaced by our own. The program then locates the new routine at the top of memory. Now each time a byte is to be examined during execution it must first pass through our checkpoint. Once the origin of the call is established, the routine looks for the CLS token, 132 (84H). Only when it has been located does the routine proceed to examine the next byte. This is checked to see if it lies in the range 0-9. Once it has passed this test, the clear screen routine is implemented after first calculating the appropriate value with which to fill the screen. You will notice that not only is it necessary to check for the new command, but also to provide the routine which implements it. In this case a simple block load to the screen has been used. Control is then returned to the ROM processing routine, which prepares to examine the byte following our new command. So, as far as the VZ knows, everything is continuing normally. Tricky isn't it?

I have already successfully used this approach to produce a VZ Printer Patch, which enables all the normal printer functions for owners of EPSON or EPSON compatible printers. The COPY command is intercepted by the patch and as a result its function has been enhanced to allow a proper dump of both the LO-RES and HI-RES screens. One further enhancement that could be explored would be an extension of Basic's SOUND command. The possibilities are limited only by imagination and memory.

```
;########################
;# ENHANCED CLS COMMAND #
;# BY LARRY TAYLOR 1986 #
;########################
;
;THIS SECTION RELOCATES
;THE PROGRAM TO THE TOP
;OF AVAILABLE MEMORY.
;
VCTR EQU  7A28H          ;SET VCTR AS 7A28H
     LD   SP,7700H       ;LOAD STACK POINTER
     LD   HL,(78B1H)     ;GET THE TOP OF MEMORY
     LD   BC,ENDP-NVCT   ;GET LENGTH OF PROGRAM
     PUSH BC             ;SAVE PROGRAM LENGTH
     XOR  A              ;RESET ALL FLAGS
     SBC  HL,BC          ;TAKE LENGTH FROM TOP OF MEMORY
     LD   (78B1H),HL     ;LOAD NEW TOP OF MEMORY
     PUSH HL             ;SAVE NEW TOP OF MEMORY
     XOR  A              ;RESET ALL FLAGS
     LD   BC,33H         ;RESERVE 50 BYTES STRING SPACE
     SBC  HL,BC          ;TAKE SPACE FROM TOP OF MEMORY
     LD   (78A0H),HL     ;LOAD START OF STRING SPACE
     POP  DE             ;RETRIEVE TOP OF MEMORY
     INC  DE             ;INCREASE BY ONE
     LD   HL,(7804H)     ;GET CURRENT RST10H VECTOR
     LD   (VCTR),HL      ;STORE IT IN 7A28H
     LD   (7804H),DE     ;LOAD NEW VECTOR
     LD   HL,NVCT        ;GET START OF PROGRAM TO MOVE
     POP  BC             ;RETRIEVE PROGRAM LENGTH
     LDIR                ;MOVE TO NEW LOCATION
     CALL 1B4DH          ;DO A NEW
     JP   1A19H          ;JUMP TO READY MESSAGE
```

```
;
;START OF THE PROCESSING
;ROUTINE FOR NEW COMMAND.
;
NVCT EXX                          ;SAVE ALL REGISTERS
     LD   HL,1D5BH                ;CHECK TO
     POP  DE                      ;SEE IF THE
     OR   A                       ;RETURN
     SBC  HL,DE                   ;ADDRESS
     PUSH DE                      ;IS 1D5BH
     EXX                          ;RESTORE ALL REGISTERS
     JP   NZ,1D78H                ;IF NOT GO TO NORMAL PROCESSING
     PUSH HL                      ;SAVE STRING ADDRESS
     CALL 1D78H                   ;GET NEXT VALUE FROM STRING
     JR   NZ,CONT                 ;IF NOT ZERO THEN CONTINUE
POP  POP  HL                      ;ELSE RESTORE STRING ADDRESS
     LD   DE,(VCTR)               ;RETRIEVE ORIGINAL VECTOR
     PUSH DE                      ;AND JUMP
     RET                          ;TO IT
CONT CP   84H                     ;CHECK FOR CLS TOKEN
     JR   NZ,POP                  ;IF NOT FOUND RETURN TO CALLER
     INC  HL                      ;MOVE TO NEXT VALUE IN STRING
     LD   A,(HL)                  ;GET NEXT VALUE AFTER CLS TOKEN
     SUB  30H                     ;REDUCE IT TO RANGE 0-8
     JR   Z,EXEC                  ;IF ZERO THEN EXECUTE COMMAND
     LD   B,8                     ;LOAD B REG WITH UPPER LIMIT
CMPR CP   B                       ;CHECK IF A=B
     JR   Z,EXEC                  ;IF YES THEN EXECUTE COMMAND
     DJNZ CMPR                    ;REDUCE B AND CONTINUE CHECK
     JR   POP                     ;NO MATCH SO RETURN TO CALLER
EXEC POP  DE                      ;RETRIEVE OLD STRING ADDRESS
     POP  DE                      ;RETRIEVE OLD RETURN ADDRESS
     LD   DE,1D1EH                ;LOAD NEW RETURN ADDRESS
     PUSH DE                      ;SAVE NEW RETURN ADDRESS
     INC  HL                      ;MOVE TO NEXT VALUE IN STRING
     PUSH HL                      ;SAVE CURRENT STRING ADDRESS
     ADD  A,A                     ;MULTIPLY CLS
     ADD  A,A                     ;VALUE BY 16 TO
     ADD  A,A                     ;CALCULATE THE
     ADD  A,A                     ;COLOUR OFFSET
     JR   NZ,SKIP                 ;IF RESULT NOT ZERO THEN SKIP
     INC  A                       ;IF ZERO INCREASE TO ONE
SKIP ADD  A,7FH                   ;ADD 127 TO GET GRAPHICS BLOCK
;
;CLEAR SCREEN ROUTINE
;
     LD   HL,7000H                ;LOAD START OF SCREEN ADDRESS
     LD   (7820H),HL              ;SET CURSOR POSITION
     LD   DE,7001H                ;LOAD START OF SCREEN PLUS ONE
     LD   BC,01FFH                ;NUMBER OF BYTES TO MOVE
     LD   (HL),A                  ;LOAD GRAPHICS BLOCK INTO HL
     LDIR                         ;DO A BLOCK FILL OF THE SCREEN
     POP  HL                      ;RETRIEVE STRING ADDRESS
     RET                          ;RETURN TO 1D1EH TO CONTINUE
ENDP DEFB 0                       ;END OF PROGRAM MARKER
```

COURTESY D.SMITH ELECTRONICS - 20

& HUNTER VALLEY VZ USERS' GROUP

